A Brief Introduction to Mixup Method

Haozhe Feng

State Key Lab of CAD& CG, Real Doctor AI Research Centre

fenghz@zju.edu.cn

June 14, 2019

Outline

- The Insight of Mixup Method: from Emperical Risk Minimization(ERM) to Vicincal Risk Minimization(VRM)
- Input Mixup and Manifold Mixup
- The Utilization of Mixup Method

The Insight of Mixup Method: from ERM to VRM

The learning problem can be formulated as the search of the function $f \in \mathcal{F}$ that minimizes the expectation of the given loss $l(f(x), y)$

$$R(f) = \int l(f(x), y) dP(x, y) \tag{1}$$

- Emperical Risk Minimization(ERM)

$$dP_{emp}(x, y) = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}(x) \delta_{y_i}(y) \tag{2}$$

Using P_{emp}, we can approximate $R(f)$ by the empirical risk

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i) \tag{3}$$

However, the empirical risk (3) monitors the behaviour of f only at a finte set of n examples. When considering the universarial approximation theorem, one trival way to minimize (3) is to memorize the training data[1], which leads to the undesirable behaviour of f outside the training data[2].

The Insight of Mixup Method: from ERM to VRM

- Vicincal Risk Minimization(VRM)

$$dP_{VRM}(x,y) = \frac{1}{n}\sum_{i=1}^{n} dP_{x_i}(x)\delta_{y_i}(y) \tag{4}$$

Here we usually set $dP_{x_i}(x)$ as spherical gaussian kernel functions $\mathcal{N}(x_i, \delta^2)$. To learn with VRM, we can construct a dataset $\mathcal{D}_v = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{m}$ with $(\tilde{x}_i, \tilde{y}_i) \sim dP_{VRM}(x,y)$ and then utilize ERM in $\mathcal{D}_v$ as following

$$R_{VRM}(f) = \frac{1}{m}\sum_{i=1}^{m} l(f(\tilde{x}_i), \tilde{y}_i) \tag{5}$$

But it still doesn't give a natural form of the distribution of $p_{y_i}(y)$ and the direct function $\delta_{y_i}(y)$ is non-convex and rigorous.

The Insight of Mixup Method: from ERM to VRM

- The Mixup Form of $dP(x, y)$ The paper propose a generic vicinal distribution called mixup:

$$\mu(\tilde{x}, \tilde{y}|x_i, y_i) = \frac{1}{n}\sum_{j=1}^{n} E_{\lambda \sim Beta(\alpha,\alpha), \alpha \in (0,\infty)}[\delta(\tilde{x} = \lambda x_i + (1-\lambda)x_j, \tilde{y} = \lambda y_i + (1-\lambda)y_j)] \quad (6)$$

In a nutshell, sampling process from $dP(x, y)$ with the form of (6) can be written as

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j \quad (7)$$

$$\tilde{y} = \lambda y_i + (1-\lambda)y_j \quad (8)$$

where (x_i, y_i) and (x_j, y_j) are two pairs randomly choosed from the training data and λ is sampled from $Beta(\alpha, \alpha)$.

Input Mixup

Utilizing the input mixup trick $(8) - (10)$ can directly raise the classify accuracy, just as the following tables show

Model	Method	Epochs	Top-1 Error	Top-5 Error
ResNet-50	ERM (Goyal et al., 2017)	90	23.5	-
	mixup $\alpha = 0.2$	90	**23.3**	**6.6**
ResNet-101	ERM (Goyal et al., 2017)	90	22.1	-
	mixup $\alpha = 0.2$	90	**21.5**	**5.6**
ResNeXt-101 32*4d	ERM (Xie et al., 2016)	100	21.2	-
	ERM	90	21.2	5.6
	mixup $\alpha = 0.4$	90	**20.7**	**5.3**
ResNeXt-101 64*4d	ERM (Xie et al., 2016)	100	20.4	5.3
	mixup $\alpha = 0.4$	90	**19.8**	**4.9**
ResNet-50	ERM	200	23.6	7.0
	mixup $\alpha = 0.2$	200	**22.1**	**6.1**
ResNet-101	ERM	200	22.0	6.1
	mixup $\alpha = 0.2$	200	**20.8**	**5.4**
ResNeXt-101 32*4d	ERM	200	21.3	5.9
	mixup $\alpha = 0.4$	200	**20.1**	**5.0**

Table 1: Validation errors for ERM and *mixup* on the development set of ImageNet-2012.

Method	Specification	Modified		Weight decay	
		Input	Target	10^{-4}	5×10^{-4}
ERM		✗	✗	5.53	5.18
mixup	AC + RP	✓	✓	**4.24**	4.68
	AC + KNN	✓	✓	4.98	5.26
mix labels and latent representations (AC + RP)	Layer 1	✓	✓	4.44	**4.51**
	Layer 2	✓	✓	4.56	4.61
	Layer 3	✓	✓	5.39	5.55
	Layer 4	✓	✓	5.95	5.43
	Layer 5	✓	✓	5.39	5.15

Table 5: Results of the ablation studies on the CIFAR-10 dataset. Reported are the median test errors of the last 10 epochs. A tick (✓) means the component is different from standard ERM training, whereas a cross (✗) means it follows the standard training practice. AC: mix between all classes. SC: mix within the same class. RP: mix between random pairs. KNN: mix between k-nearest neighbors (k=200). Please refer to the text for details about the experiments and interpretations.

We can see that in [3], mixup only works in (input space, label space). But this deduction is very anti-intuitive, for the arithmetic operations attribute in latent space have shown some linear properties[4].

Manifold Mixup

The manifold mixup loss can be discribed as following:

1. For each minibatch, select a random layer k which split the forward network into two parts: $N(x) = f_k g_k(x)$ and sample $\lambda \sim Beta(\alpha, \alpha)$
2. Minimize the manifold mixup loss

$$L = E_{(x_i,y_i),(x_j,y_j)\in \text{ minibatch}} l(f_k(\lambda g_k(x_i) + (1-\lambda)g_k(x_j)), \lambda y_i + (1-\lambda)y_j) \tag{9}$$

[5] can seen as a theoretical complement for [3]. It explains how manifold mixup change the learned representations as well as gives the condition that manifold mixup works better than input mixup.

Manifold Mixup

Model	Test Error	Test NLL
PreActResNet18		
No Mixup	5.12	0.2646
Input Mixup ($\alpha = 1.0$) †	3.90	n/a
AdaMix ‡	3.52	n/a
Input Mixup ($\alpha = 1.0$)	3.50	0.1945
Manifold Mixup ($\alpha = 2.0$)	**2.89**	**0.1407**
PreActResNet152		
No Mixup	4.20	0.1994
Input Mixup ($\alpha = 1.0$)	3.15	0.2312
Manifold Mixup ($\alpha = 2.0$)	2.76	0.1419
Manifold Mixup all layers ($\alpha = 6.0$)	**2.38**	**0.0957**

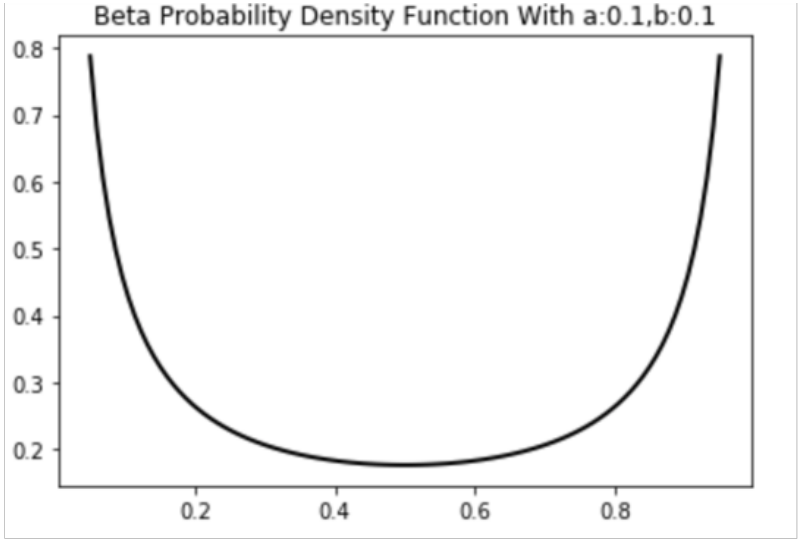
(a) CIFAR-10

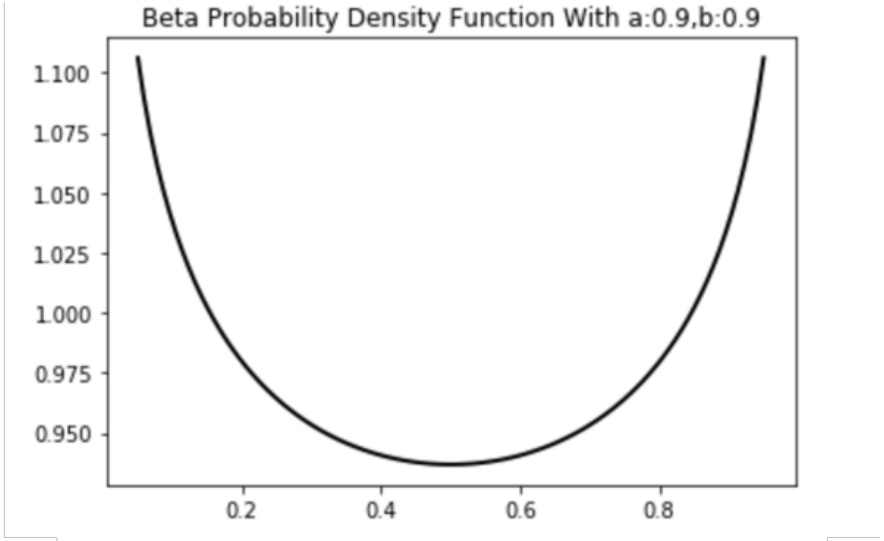
Model	Test Error	Test NLL
PreActResNet18		
No Mixup †	25.60	n/a
No Mixup	24.68	1.284
Input Mixup ($\alpha = 1.0$) †	21.10	n/a
Manifold Mixup ($\alpha = 2.0$)	**21.05**	**0.913**
PreActResNet34		
Input Mixup ($\alpha = 1.0$)	22.79	1.085
Manifold Mixup ($\alpha = 2.0$)	**20.39**	**0.930**

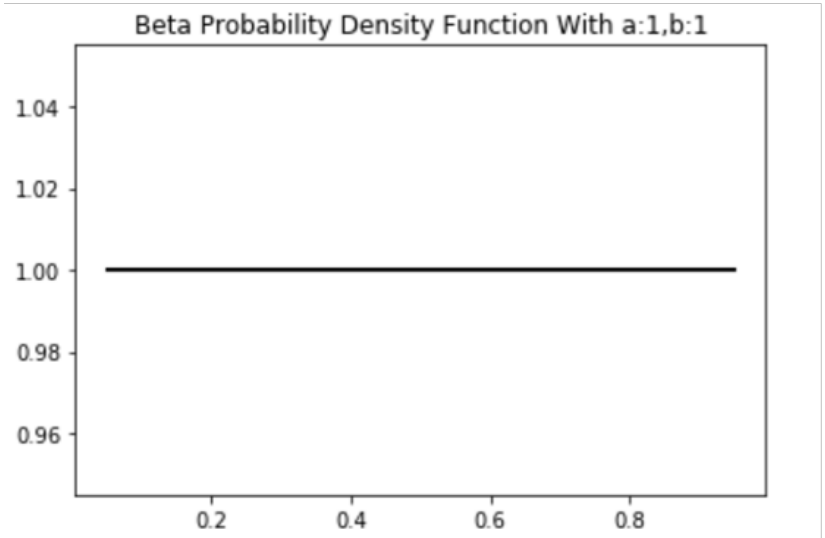
(b) CIFAR-100

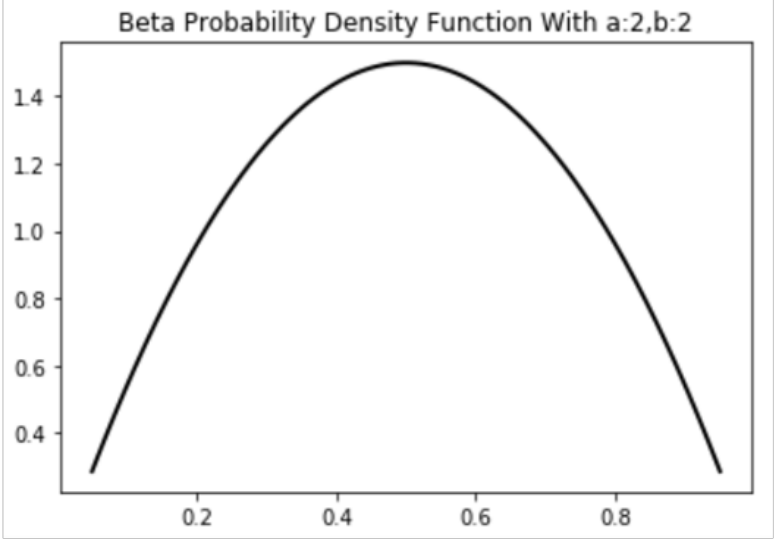
The main difference between table 1,2 and table 3 is the choose of parameter for beta distribution.

Manifold Mixup









The Utilization of Mixup Method

- Regularization on Supervised learning
  Just as table 3 shows. Besides this, the authors performed an experiment where they trained with Manifold Mixup but blocked gradients immediately after the layer where they perform mixup(fronzen the parameters of f_k). The results(4.33) is better than the baseline(5.12), but worse than the both manifold and input mixup methods. This demonstrates that the Manifold Mixup method improves results by changing the layers both before and after the mixup operation is applied.

The Utilization of Mixup Method

- Robust to corrupted labels
  [1] points out that the network just "memorizes" the label. The two assumption of robustion is that
  1. Increasing the strength of mixup interpolation α should generate virtual examples further from the training examples, making the memorization more difficult to achieve.
  2. Learn interpolations between real examples is much easier than memorizing interpolations involving random labels.

The Utilization of Mixup Method

- Robustness to corrupted labels

Label corruption	Method	Test error		Training error	
		Best	Last	Real	Corrupted
20%	ERM	12.7	16.6	0.05	0.28
	ERM + dropout ($p = 0.7$)	8.8	10.4	5.26	83.55
	mixup ($\alpha = 8$)	**5.9**	6.4	2.27	86.32
	mixup + dropout ($\alpha = 4, p = 0.1$)	6.2	**6.2**	1.92	85.02
50%	ERM	18.8	44.6	0.26	0.64
	ERM + dropout ($p = 0.8$)	14.1	15.5	12.71	86.98
	mixup ($\alpha = 32$)	11.3	12.7	5.84	85.71
	mixup + dropout ($\alpha = 8, p = 0.3$)	**10.9**	**10.9**	7.56	87.90
80%	ERM	36.5	73.9	0.62	0.83
	ERM + dropout ($p = 0.8$)	30.9	35.1	29.84	86.37
	mixup ($\alpha = 32$)	25.3	30.9	18.92	85.44
	mixup + dropout ($\alpha = 8, p = 0.3$)	**24.0**	**24.8**	19.70	87.67

The Utilization of Mixup Method

- Robustness to Adversarial Examples

Metric	Method	FGSM	I-FGSM
Top-1	ERM	90.7	99.9
	mixup	**75.2**	99.6
Top-5	ERM	63.1	93.4
	mixup	**49.1**	95.8

(a) White box attacks.

Metric	Method	FGSM	I-FGSM
Top-1	ERM	57.0	57.3
	mixup	**46.0**	**40.9**
Top-5	ERM	24.8	18.1
	mixup	**17.4**	**11.8**

(b) Black box attacks.

The Utilization of Mixup Method

- Semi-supervised learning
  We use the loss function in [6] to explain the role of mixup in semi-supervised learning

$$u_i, u_j \sim \mathcal{U} \tag{10}$$
$$q_i, q_j = model(u_i), model(u_j) \tag{11}$$
$$\lambda \sim beta(\alpha, \alpha) \tag{12}$$
$$u' = \lambda u_i + (1-\lambda) u_j \tag{13}$$
$$q' = \lambda q_i + (1-\lambda) q_j \tag{14}$$
$$Loss_{\mathcal{U}} = \|q' - model(u')\|_2^2 \tag{15}$$

均匀设计与正交设计 [7]

假设我们有

- s 个超参数需要调节
  e.g. Learning Rate, Model Structure, Data Augmentation
- 每个超参数有 q 个水平
  e.g. Learning Rate:[1e-3,1e-4,1e-5],Model Structure:[Resnet,Dense121,PreActResnet18],Data Augmentation Prob:[0.1,0.5,0.7]

如果要用网格法全部试一次需要 q^s 次实验。
实验设计的目的是尽量让实验在参数空间中均匀分布，在只进行少数次实验时就能找到靠谱的超参数。好的实验设计在挑选代表实验策略的时候要做到" 均匀分散，整齐可比"。为了保持整齐可比的特点，正交设计提出了 q^2 次实验设计方法，如果我们只需要做到" 均匀分散"，那么均匀设计提出了 q 次实验设计方法，这种设计方法利用了数论中质数的均匀分布规律。在使用过程中，我们不用过分关注其数学推理，只需要关注如何使用。这里给出两个例子

均匀设计与正交设计 [7]

表2 正交表 $L_9(3^4)$

No.	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	2	2
8	3	2	1	3
9	3	3	2	1

表6 $U_7^*(7^4)$

	1	2	3	4
1	1	3	5	7
2	2	6	2	6
3	3	1	7	5
4	4	4	4	4
5	5	7	1	3
6	6	2	6	2
7	7	5	3	1


Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals.
Understanding deep learning requires rethinking generalization.
arXiv preprint arXiv:1611.03530, 2016.


Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus.
Intriguing properties of neural networks.
arXiv preprint arXiv:1312.6199, 2013.


Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz.
mixup: Beyond empirical risk minimization.
arXiv preprint arXiv:1710.09412, 2017.


Tom White.
Sampling generative networks.
arXiv preprint arXiv:1609.04468, 2016.


Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio.

Manifold mixup: Better representations by interpolating hidden states.

In *International Conference on Machine Learning*, pages 6438–6447, 2019.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel.
Mixmatch: A holistic approach to semi-supervised learning.
arXiv preprint arXiv:1905.02249, 2019.

方开泰.
均匀设计及其应用.
PhD thesis, 1994.